MAPS: A Free Medical Image Processing Pipeline

Blake Lucas, Bennett Landman, Jerry Prince, Dzung Pham

Image Analysis and Communications Laboratory (IACL) & Laboratory for Medical Image Computing (MedIC),

Dept. of Electrical and Computer Engineering, The Johns Hopkins University, Baltimore, MD

Overview

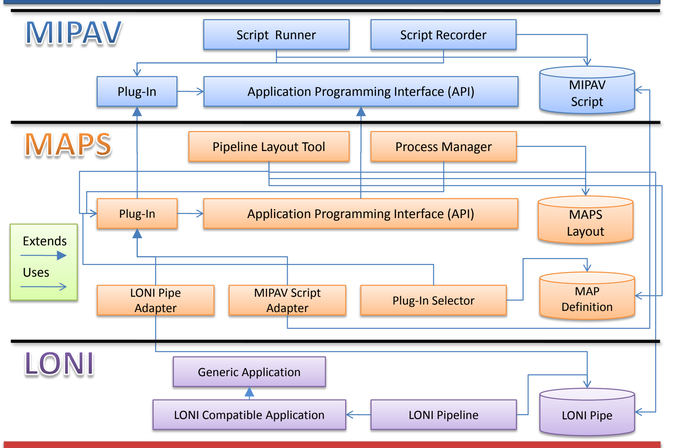
Many research-oriented image processing tasks incorporate large datasets, require multiple steps, and run software created by various institutions. Automating the execution of these multi-stage processing tasks, or pipelines, is difficult because of the complexities associated with integrating different software packages and resolving dependencies between processing steps. The **MedIC Automated Pipeline Scheduler (MAPS)** is a **user-friendly automation solution** that integrates existing automation tools, including the LONI Pipeline and MIPAV.

The **LONI Pipeline [1]** is a software application for designing pipelines built from generic applications. However, each application must first be wrapped so that it is compatible with LONI's format for program execution. Even then, only programs that support the same data formats can forward data between each other. Users who want to access LONI's library of compatible software are also required to pay a fee to process their data on LONI's servers, making the LONI Pipeline by itself an expensive solution.

One alternative is to use **MIPAV (Medical Image Processing, Analysis, and Visualization) [2]**, a free medical image analysis tool from the NIH. MIPAV has an extensive library of image processing tools, and supports over sixty different medical image formats, including DICOM. Users can develop and install new plug-ins for MIPAV, and then record macro scripts to automate execution of multiple processing tasks. However, the macro style of scripting requires the user to manually execute the pipeline at least once before the pipeline can be automated, which can be cumbersome if the pipeline takes several hours or days to run.

MAPS is an open source automation solution that integrates with the LONI Pipeline and MIPAV. It includes an intuitive **pipeline layout tool** for designing pipelines, a **process manager** to spool processing tasks over multiple processors, and an **Application Programming Interface (API)** that extends MIPAV's API for developing image processing algorithms. MAPS can **read and execute both LONI pipes and MIPAV scripts** in addition to its own plug-in format . The MAPS software package is distributed as a platform-independent Java application, publicly available from **http://maps4mipav.sourceforge.net.**

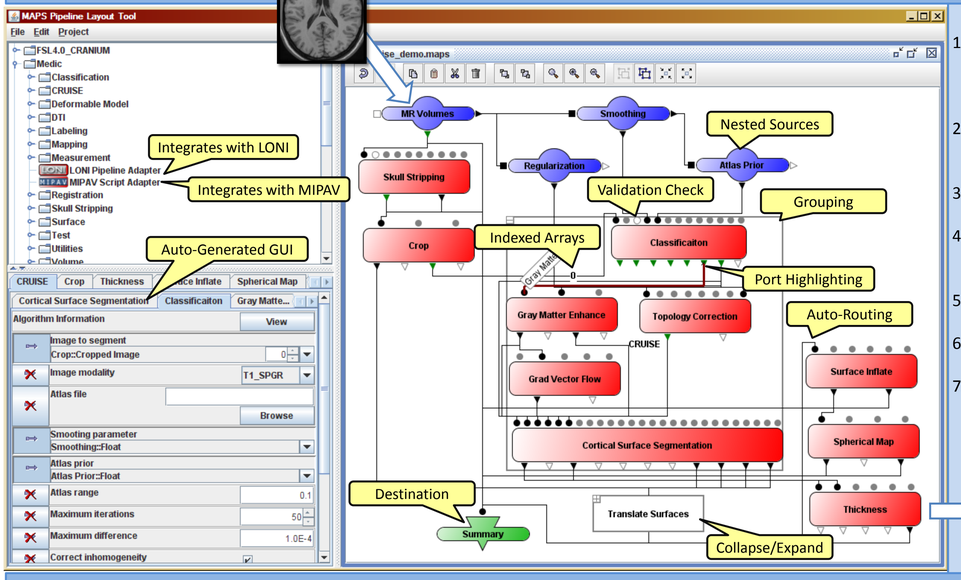
Architecture



http://maps4mipav.sourceforge.net

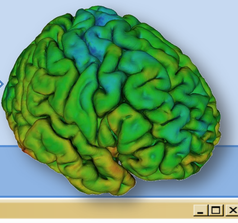
Case Example: Cortical Reconstruction

Pipeline Layout Tool



Features

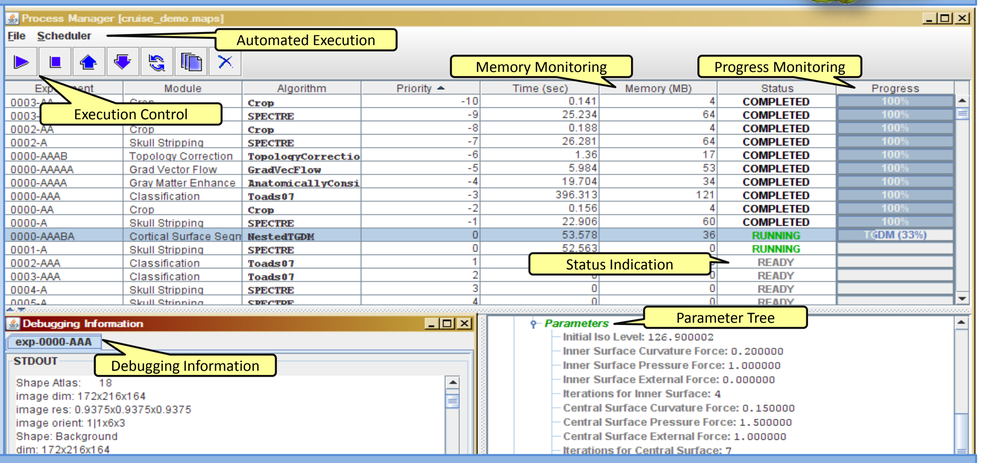
1. New plug-ins are automatically discovered and installed, so there is no need for users to define the input and output to each algorithm.
2. All plug-ins can read and write in over sixty MIPAV supported file formats.
3. Parameter sources can be nested to create "foreach" loops.
4. Destinations can summarize output data in a spreadsheet or copy the data to a specified location.
5. Modules can be grouped together and then saved to the library.
6. LONI pipe files can be opened and installed in the MAPS library.
7. MIPAV scripts can be imported as MAPS algorithms.



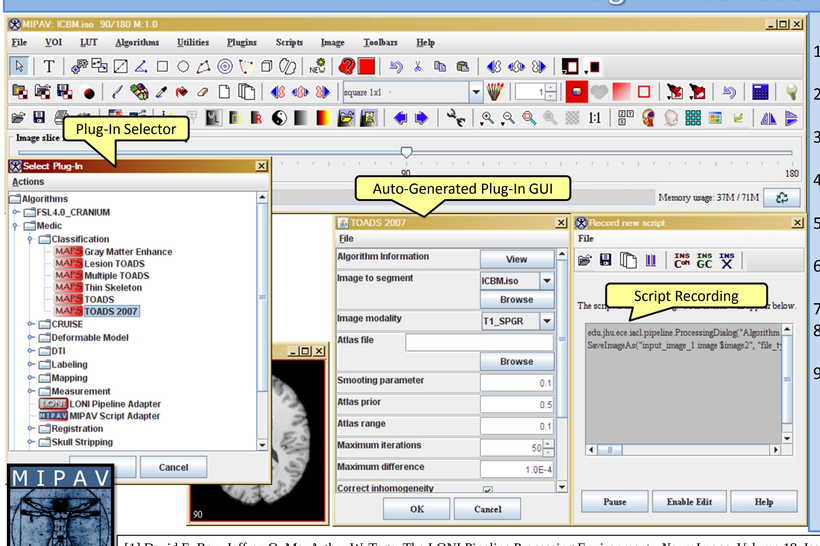
Process Manager

Features

1. All plug-ins run in a virtual MIPAV environment with access to all MIPAV's functionality.
2. The automated scheduler queues and executes multiple processes simultaneously.
3. Processes can be manually executed from the execution control toolbar.
4. Parameter information is displayed in the parameter tree.
5. Progress, status, and memory information is displayed in the process table.
6. Debugging information can be inspected during execution.
7. Execution priority can be modified.



MIPAV Plug-In Interface



Features

1. Any plug-in from the MAPS library can be executed within MIPAV.
2. MAPS plug-ins behave like ordinary MIPAV image processing plug-ins.
3. Plug-in GUIs are automatically generated based on the input output parameters to each algorithm.
4. MAPS plug-ins are recordable with MIPAV's scripting interface.
5. An XML description of the current plug-in parameters can be saved and opened.
6. Parameters are automatically validated before running each plug-in.
7. Information about a plug-in can be viewed and edited.
8. Runtime performance is monitored and recorded for all plug-ins.
9. All plug-ins can read and write in any of MIPAV's supported file formats.

[1] David E. Rex, Jeffrey Q. Ma, Arthur W. Toga, The LONI Pipeline Processing Environment, *NeuroImage*. Volume 19, Issue 3, July 2003, Pages 1033-1048.

[2] McAuliffe, M. J., Lalonde, F. M., McGarry, D., Gandler, W., Csaky, K., and Trus, B. L. 2001. Medical Image Processing, Analysis & Visualization in Clinical Research. In *IEEE Proc. Symposium on Computer-Based Medical Systems* (March 26 - 27, 2001). CBMS. IEEE Computer Society, Washington, DC, 381.

HBM 2008

14th Annual Meeting of the Organization for Human Brain Mapping